Stat Biosci (2012) 4:197–212
DOI 10.1007/s12561-011-9045-2

Stepwise Paring down Variation for Identifying Influential Multi-factor Interactions Related to a Continuous Response Variable

Jing-Shiang Hwang · Tsuey-Hwa Hu

Received: 4 November 2010 / Accepted: 26 October 2011 / Published online: 16 November 2011

Abstract Although several model-based methods are promising for the identification of influential single factors and multi-factor interactions, few are widely used in real applications for most of the model-selection procedures are complex and/or infeasible in computation for high-dimensional data. In particular, the ability of the methods to reveal more true factors and fewer false ones often relies heavily on the selection of appropriate values of tuning parameters, which is still a difficult task to practical analysts. This article provides a simple algorithm modified from stepwise forward regression for the identification of influential factors. Instead of keeping the identified factors in the next models for adjustment in stepwise regression, we propose to subtract the effects of identified factors in each run and always fit a single-term model to the effect-subtracted responses. The computation is lighter as the proposed method only involves calculations of a simple test statistic; and therefore it could be applied to screen ultrahigh-dimensional data for important single factors and multi-factor interactions. Most importantly, we have proposed a novel stopping rule of using a constant threshold for the simple test statistic, which is different from the conventional stepwise regression with AIC or BIC criterion. The performance of the new algorithm has been confirmed competitive by extensive simulation studies compared to several methods available in R packages, including the popular group lasso, surely independence screening, Bayesian quantitative trait locus mapping methods and others. Findings from two real data examples, including a genome-wide association study, demonstrate additional useful information of high-order interactions that can be gained from implementing the proposed algorithm.

J.-S. Hwang (✉) · T.-H. Hu
Institute of Statistical Science, Academia Sinica, Taipei 11529, Taiwan
e-mail: hwang@sinica.edu.tw

T.-H. Hu
e-mail: jshwang6@stat.sinica.edu.tw

Keywords Lasso · Surely independence screening · QTL · Single nucleotide polymorphism · Variable selection · Multi-factor interaction

1 Introduction

It is believed that many common diseases or traits may be associated with multiple genetic markers and environmental factors. Numerous statistical methods have been proposed for tackling the challenging problem of identifying markers causally related to trait variables. A comprehensive review of popular statistical methods, including feature/variable selection methods, multiple testing approaches, supervised statistical models and various model-based approaches specifically on the association studies of complex diseases using data on single nucleotide polymorphisms (SNPs), can be found in Liang and Kelemen [10]. The major common objective of these methods is to identify as many true influential markers as possible. Unfortunately, there is always the possibility that some false markers or "impostors" will accompany the identified influential ones [4]. Popular approaches for screening potential markers such as the lasso method can be regarded as a specific form of penalized likelihood for which theoretical properties have been intensively studied [7]. However, the difficulty of choosing a proper penalty function has prevented these approaches from being widely used in real world.

In practice, biomedical researchers have often adopted the simpler approach of testing each marker one by one against a predetermined threshold, say 10^{-7}, to screen for a manageable number of potential markers in the first stage. It is then relatively feasible to genotype the selected markers on a larger independent sample in the second stage, and to fit a model with interaction terms for the selected markers if the number is not too large [16]. But, with this strategy, we tend to select only those markers with very small p-values in the first stage, a process which may miss those important multi-marker interactions with weak marginal effects. The chance of finding true interactions is therefore small. Among the non-model-based approaches, the partition–retention method proposed by Chernoff et al. [4] has demonstrated impressive power of identifying factor interactions in several real data sets (see also [11]). However, improving on currently available methods and developing new and simple screening methods with greater power for studies involving a huge number of factors is still highly desirable.

This article focuses on analyzing data with a continuous response variable and m factor variables with the objective of screening out influential single factors and multi-factor interactions. Stepwise forward regression is a potential method for screening out important single factors for its outstanding performance and simplicity. However, the stopping rule for the case $m \gg n$ is a challenging problem, and computational burden of adding all identified variables in the repeatedly model fitting process is also an obstacle to being widely applied to ultrahigh-dimensional data in practice. In this study, we proposed a screening algorithm with procedures similar to stepwise forward regression. The major difference between the proposed screening method and conventional stepwise regression is a new stopping rule which we found competitive and feasible computationally for ultrahigh-dimensional data analysis from comprehensive simulation studies and real data analysis.

The paper is organized as follows: In Sect. 2, we formalize our description of our algorithm to identify influential single factors and multi-factor interactions and expound on the idea with an illustrative example. In Sect. 3, several simulated examples are used to demonstrate the performance of the proposed algorithm in comparison with several popular methods. Two real data examples, including a genome-wide association study, are given in Sect. 4 to show significant gains made by the algorithm. We conclude with remarks on the advantages and limitations of the algorithm in Sect. 5.

2 Method

The SPV Algorithm

Instead of keeping all the identified variables in the model for searching the next important variable in stepwise forward regression, we remove the effects of the identified variables from the original responses and use the residuals as the refined responses for searching the next influential variable. Specifically, the algorithm starts with a run of fitting m single-factor ANOVA models. The largest estimated effect (or negative log p-value) among the m factors is then compared to a pre-defined threshold. If the factor with the largest effect is identified as influential, the estimated effects of the factor are then subtracted from the responses. We refer to these residuals as "refined responses." Through this process, the total variation of the original responses is pared down by the amount contributed by the identified factor. We then repeat a run of fitting m single-factor ANOVA models to the newly refined responses and again pick the factor with the largest estimated effect. If the estimated effect for this factor is larger than the threshold, the identified factor is also recorded as important and its estimated effects on the observations are further deducted from the current refined responses to form a new set of refined responses. The total variation of refined responses and estimated variance of the model error component are thus further reduced. Accordingly, the remaining undiscovered causal factors have increased chances of being identified in the following repeated processes.

If no estimated effect of single factor larger than the threshold, we move to the next stage of exploring two-factor interactions by a run of fitting each of the $C_2^m = m(m-1)/2$ ANOVA models, each with only a single term of two-factor interaction. The same procedures used for screening single factors with a new threshold are applied to search for important two-factor interactions. When this is done, the procedures can be repeated in the next stages to screen for higher multi-factor interactions.

In each run of the algorithm, the total variation of refined responses are pared down from the previous run so that remaining factors with causal effects have increased chances of being found. Hence, we term the proposed method the stepwise paring-down variation (SPV) algorithm.

Determination of the Threshold

The threshold can be determined by using a permutation approach when enough computational power is available. In screening for important d-factor interactions,

we repeatedly fit all the C_d^m single-term ANOVA models of the algorithm using a response vector consisting of random permutation of the original responses, and obtain the smallest p-value. This value represents a realization of the minimum of the C_d^m p-values when none of the C_d^m d-factor combinations is correlated with a noisy response variable. We usually need a large number of repetitions, say $B = 100$, to collect a set of such minimum values, $p_d^{(1)}, \ldots, p_d^{(B)}$. Let u_d and s_d denote, respectively, the sample mean and standard deviation of these negative log p-values, $\{-\log(p_d^{(b)})\}_1^B$. We then determine the threshold as $\tau_d = u_d + \rho_d s_d$. The value of ρ_d may be chosen according to the desired level of stringency in screening. We would suggest simply choosing $\rho_d = 0$ for a moderate threshold or $\rho_d = 2$ for a stringent one.

As the permutation method is feasible only when m is small, we need an alternative approach for determining the thresholds in practical applications. In testing each d-factor combination, if m is large and none of the d-factor combinations are causally related to the responses, the smallest p-value among the C_d^m p-values would approximate to a beta distribution with parameters 1 and C_d^m [8, 12]. Hence, the ath percentile of Beta$(1, C_d^m)$, denoted by p_d^a, can be used for determining the importance of the d-factor combination with the smallest p-value. Accordingly, we may set a threshold $\tau_d = -\log(p_d^a)$. We find that $a = 25$ is a proper choice in terms of overall performance in our simulation studies. In practice, one may set a stringent threshold by choosing a very lower value of $a = 5$ to reduce false positives or a higher $a = 50$ to avoid losing important factors and allow some unimportant factors.

The Idea

We shall use a simple ANOVA model to show the idea with which the SPV algorithm works. The basic idea comes from examining the ANOVA table. Suppose that only two of m uncorrelated factors, X_1 and X_2, have effects on the responses. In other words, the responses may be generated from the model presented as

$$y_i = \mu + \sum_{a=1}^{c_1} \beta_{1_a} I(X_{i1} = a) + \sum_{b=1}^{c_2} \beta_{2_b} I(X_{i2} = b) + \varepsilon_i,$$

where μ is a constant, $I(\cdot)$ is an indicator function, β_{1_a} is the effect of the ath of c_1 levels of the X_1 variable with constraint $\sum \beta_{1_a} = 0$, β_{2_b} is the effect of the bth of c_2 levels of the X_2 variable with constraint $\sum \beta_{2_b} = 0$, and $\varepsilon_i \sim N(0, \sigma^2)$ is a random error component. Let the average of all responses under the ath level of variable X_l be denoted by $\bar{y}_{l_a} = \sum_{i=1}^n \frac{y_i I(X_{il}=a)}{n_{l_a}}$, where $n_{l_a} = \sum_{i=1}^n I(X_{il} = a)$ is the size of the corresponding category. The total sum of squares may be written as

$$\sum_{i=1}^n (y_i - \bar{y})^2 = \sum_{i=1}^n \left[\sum_{a=1}^{c_1} \bar{y}_{1_a} I(X_{i1} = a) - \bar{y} \right]^2 + \sum_{i=1}^n \left[\sum_{b=1}^{c_2} \bar{y}_{2_b} I(X_{i2} = b) - \bar{y} \right]^2 + \sum_{i=1}^n \left[y_i - \sum_{a=1}^{c_1} \bar{y}_{1_a} I(X_{i1} = a) - \sum_{b=1}^{c_2} \bar{y}_{2_b} I(X_{i2} = b) + \bar{y} \right]^2.$$

This is usually written as $SS_T = SS_{X_1} + SS_{X_2} + SS_E$. The number of degrees of freedom associated with each respective sum of squares is $n-1$, c_1-1, c_2-1 and $n-c_1-c_2+1$.

In the first run of searching for the most influential one of the m factors, we fit a single-term ANOVA model $y_i - \bar{y} = \sum_{a=1}^{c_l} \lambda_{l_a} I(X_{il} = a) + \varepsilon_i$ to the response variable and factor X_l for $l = 1, \ldots, m$. The test statistic is

$$\mathrm{F}^*(X_l) = \frac{SS_{X_l}/(c_l-1)}{(SS_T - SS_{X_l})/(n-c_l+1)} = \frac{SS_{X_l}/(c_l-1)}{SS_{E|X_l}/(n-c_l+1)} = \frac{MSS_{X_l}}{MSS_{E|X_l}}.$$

If X_l is not one of the two causal factors, i.e. $l > 2$, the ratio between the expected mean sums of squares for the factor and expected mean squared error is

$$\frac{E\{MSS_{X_l}\}}{E\{MSS_{E|X_l}\}} = \frac{\sigma^2 + \frac{\sum_{a=1}^{c_l} n_{l_a} \lambda_{l_a}^2}{c_l - 1}}{\sigma^2 + \frac{\sum_{a=1}^{c_1} n_{1_a} \beta_{1_a}^2}{c_1-1} + \frac{\sum_{b=1}^{c_2} n_{2_b} \beta_{2_b}^2}{c_2-1}}.$$

We expect that the $-\log$ p-value for this $\mathrm{F}^*(X_l)$ statistic has little chance of being larger than the threshold level because the mean effect size of X_l, $\sum_{a=1}^{c_l} n_{l_a} \lambda_{l_a}^2$, is zero.

On the other hand, when X_1 or X_2 is considered in the single-term ANOVA model, the two mean sum of squares ratios are

$$\frac{E\{MSS_{X_1}\}}{E\{MSS_{E|X_1}\}} = \frac{\sigma^2 + \frac{\sum_{a=1}^{c_1} n_{1_a} \beta_{1_a}^2}{c_1-1}}{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b} \beta_{2_b}^2}{c_2-1}} \quad \text{and} \quad \frac{E\{MSS_{X_2}\}}{E\{MSS_{E|X_2}\}} = \frac{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b} \beta_{2_b}^2}{c_2-1}}{\sigma^2 + \frac{\sum_{a=1}^{c_1} n_{1_a} \beta_{1_a}^2}{c_1-1}}.$$

Whether the p-values of the F^* statistics for factor X_1 and X_2 are significant depend not only on the mean effect size of each factor but also the mean effect size of the other factor. This fact implies that there is no guarantee that both of the two factors are significant if we run separate single-term models for each factor. In fact, it is highly possible that only one of the two is identified. For getting an insight, we simulated 500 data sets with nonzero coefficients only for the first two factors, $\beta_1 = (1.5, 0, -1.5)$ and $\beta_2 = (0.4, 0, -0.4)$, and $\sigma = 1$. All the F values for testing X_1 in the 500 simulated data sets are clearly far larger than the cut-point of significance level 0.05. Figure 1 shows that only about 8% of the F values for testing X_2 are larger than the cut-point of significance level 0.05.

The SPV algorithm starts by comparing the smallest p-value in the first run with a threshold level. If the mean effect size of X_1 is larger than the mean effect size of X_2, we expect that factor X_1 will be identified for its negative log p-value being the largest and it is also very likely to be larger than the threshold in the first run. The factor effect is then removed from the responses to form refined responses

$$y_i^{(2)} = y_i - \sum_{a=1}^{c_1} \bar{y}_{1_a} I(X_{i1} = a) \quad \text{for } i = 1, \ldots, n.$$

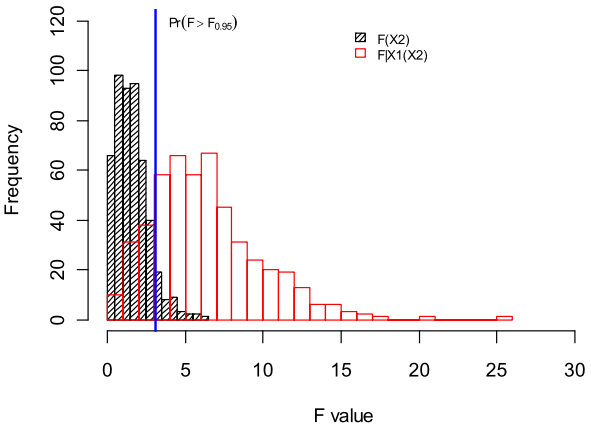

Fig. 1 Distributions of F values for testing X_2 based on 500 simulated data sets. *The shaded histogram* shows distribution of $F(X_2)$, the results of the using original responses; while *the empty histogram* is for $F|X_1(X_2)$, the results of using the refined responses with the effects of X_1 subtracted from the original responses

Accordingly, the total sum of squares is pared down to

$$\sum_{i=1}^{n}\left[y_i^{(2)}\right]^2 = \sum_{i=1}^{n}\left[\sum_{b=1}^{c_2}\bar{y}_{2_b} I\left(X_{i2}=b\right)-\bar{y}\right]^2 + \sum_{i=1}^{n}\left[y_i^{(2)} - \sum_{b=1}^{c_2}\bar{y}_{2_b} I\left(X_{i2}=b\right)+\bar{y}\right]^2,$$

or denoted by $SS_{T|X_1} = SS_{X_2} + SS_{E|X_1}$.

In the second run of the single-term ANOVA model with the effects of X_1 having been deducted, if X_l is not the causal factor X_2, the ratio between the expected mean sums of squares for the factor and expected mean squared error is

$$\frac{E\{MSS_{X_l}\}}{E\{MSS_{E|X_1,X_l}\}} = \frac{\sigma^2 + \frac{\sum_{a=1}^{c_l} n_{l_a}\lambda_{l_a}^2}{c_l-1}}{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b}\beta_{2_b}^2}{c_2-1}} = \frac{\sigma^2}{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b}\beta_{2_b}^2}{c_2-1}},$$

where $MSS_{E|X_1,X_l} = (SS_{T|X_1} - SS_{X_l})/(n-c_l-1)$. Again, we expect that the $-\log$ p-value for this $F^*(X_l)$ statistic has little chance of being larger than the threshold level for $\sum_{a=1}^{c_l} n_{l_a}\lambda_{l_a}^2 = 0$. On the other hand, we expect that variable X_2 has an increased chance of being identified as an influential factor in this run because the updated ratio between the mean sum of squares for factor X_2 and model error variance is larger than that in the previous run and also larger than those of unimportant factors in this run, that is,

$$\frac{E\{MSS_{X_2}\}}{E\{MSS_{E|X_1,X_2}\}} = \frac{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b}\beta_{2_b}^2}{c_2-1}}{\sigma^2} > \frac{\sigma^2}{\sigma^2 + \frac{\sum_{b=1}^{c_2} n_{2_b}\beta_{2_b}^2}{c_2-1}}$$

$$= \frac{E\{MSS_{X_l}\}}{E\{MSS_{E|X_1,X_l}\}} \quad \text{for } l \neq 2.$$

Therefore, we expect a smaller p-value for factor X_2 now. In the 500 simulated data sets, F values of testing X_2 after subtracting effects of X_1 shown in Fig. 1 indicate that the chance of being identified at the second run of SPV has been increased to 83%.

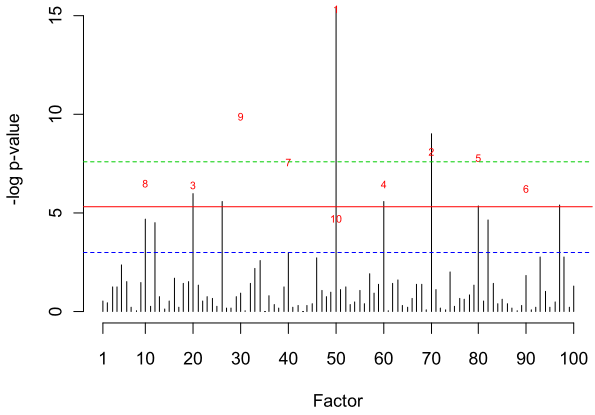

Fig. 2 The $-\log$ p-values obtained from 10 runs of the single-factor SPV algorithm on a simulated data set with each numerical symbol indicating the order and factor identified. *The solid line* is the threshold level for SPV, and *the vertical bars* indicate $-\log$ p-values of the statistics for testing each single factor. *The two dotted lines* are two threshold levels of $-\log(0.0005)$ and $-\log(0.05)$

If factor X_2 is successfully identified, the newly refined responses are formed by removing the factor effects. The mean sum of squares of the newly refined responses is further cut down to the variance of the model error component. Since the ratio between the mean sum of squares for any factor and the error variance is close to one, the chance of any single variable to be found from fitting a one-single-factor ANOVA model on these refined responses is therefore very small.

An Illustrative Example

To provide insight into how the total variations of responses are pared down and influential factors are screened out in the process, we applied the SPV algorithm to a simulated data set and compared the results with a conventional multiple test using Bonferroni correction. We generated $m = 100$ independent factors of three levels 0, 1, 2 with equal probabilities for $n = 100$ subjects. The responses were affected by the nine single factors $X_{10}, X_{20}, \ldots, X_{90}$ through the model,

$$y_i = \sum_{j\in\{10,20,\ldots,90\}} \{0.75 \times I(X_{ij} = 1) - 0.75 \times I(X_{ij} = 0)\} + \varepsilon_i,$$

where ε_i was a standard normal distribution. Each of the nine factors was supposed to contribute the same mean effect. But given a simulated data set, the estimated factor effects may not be close to the expected ones, especially when sample size is small. Accordingly, there is no guarantee of the identification of the nine important factors if we test each factor separately, as illustrated in Fig. 2.

The vertical lines in Fig. 2 show the $-\log$ p-values associated with testing each individual factor on a simulated data set, which are also the 100 $-\log$ p-values in the first run of the SPV algorithm. Only factors X_{50} and X_{70} were significantly identified when we used the most conservative Bonferroni correction level of $-\log(0.0005)$, the top dotted horizontal line in the plot. Six true factors and four false ones were revealed if we used the other extreme level of $-\log(0.05)$, the bottom dotted line. The SPV algorithm correctly identified the nine factors in the first nine runs; see the $-\log$ p-values marked with the order of identified factor in Fig. 3. The SPV algorithm

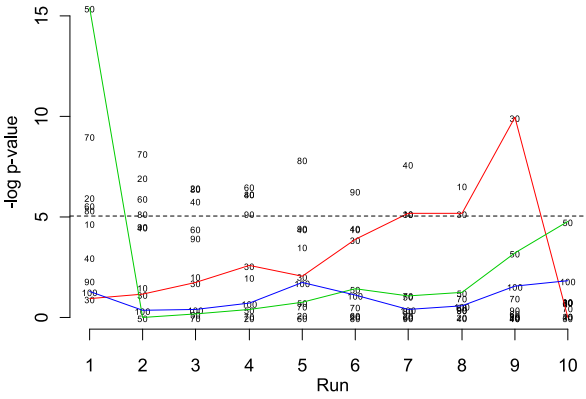

Fig. 3 The $-\log$ p-values for the nine causally related factors and an unrelated factor X_{100} marked with numeric labels in the first 10 runs of the single-factor SPV algorithm on the same simulated data set used in Fig. 2. *The dotted line* is the threshold for the SPV algorithm

stopped at the tenth run because the $-\log$ p-value corresponding to factor X_{50} was smaller than the threshold using the permutation approach with $\rho_1 = 0$, the solid horizontal line in Fig. 3.

Figure 3 shows how the nine causal factors were revealed one by one. In the first run, we see that X_{50} had the largest $-\log$ p-value, which was much larger than the threshold level, the dotted horizontal line. In the second run, the factor effect of X_{50} having been removed, its $-\log$ p-value turned out to be the smallest of the ten factors, and X_{70} was identified for having the largest $-\log$ p-value. In the third run, the $-\log$ p-values of the two identified factors were both very small and X_{20} was revealed, while the values of the others increased or changed little. For example, X_{30}, the factor with the smallest $-\log$ p-value in the first run, gradually increased to become significantly large after effects of the other eight causal factors had been removed. Unlike the nine causal factors, however, the $-\log$ p-values of the unimportant factor X_{100} had no chance of reaching significance in these ten runs.

3 Simulation Studies

The aim of these simulation studies is to evaluate the performance of the SPV algorithm compared to some available competing methods. The simulation schemes are slight modifications of those in the related literature so that direct comparison of simulation results is relatively fair.

Example 1 The first example evaluates the performance of the SPV algorithm in comparison with the group lasso variable selection approach developed by Yuan and Lin [18] and the Sure Independence Screening (SIS) by Fan and Lv [6]. We slightly modified Yuan and Lin's simulation scheme and used a much larger number of factors, as follows. In each run we first generated m-dimensional multivariate normal variables $Z_1, \ldots, Z_m$ with mean zero vector and covariance matrix $\Sigma_{ij} = 0.5^{|i-j|}$ for $m = 200$ and 1000. Subsequently, each continuous variable Z_j was transformed to a discrete variable with three levels 0, 1 or 2, such that values less than $\Phi^{-1}(1/3)$ were recoded as 0, values between $\Phi^{-1}(1/3)$ and $\Phi^{-1}(2/3)$ were recoded as 1, and values larger than $\Phi^{-1}(2/3)$ were recoded as 2. We first consider A, a set of 20 randomly

Table 1 The percentage of the average number of correctly and falsely identified factors based on 200 simulated data sets from the models of 20 causal factors in Example 1, comparing the group lasso method and SPV for sample size 300 and factor number m, and two sets of causal factors, correlated and uncorrelated

	m	Number of correct factors				Number of false factors			
		$\text{Glasso}_{0.5}$	$\text{Glasso}_{0.25}$	SIS	SPV	$\text{Glasso}_{0.5}$	$\text{Glasso}_{0.25}$	SIS	SPV
Independent factors	200	11.80	19.20	19.40	19.93	0.15	2.47	2.75	0.63
	1000	12.24	19.12	13.23	19.78	0.65	9.72	1.57	0.73
Correlated factors	200	10.97	17.93	19.41	19.30	0.03	1.38	2.35	0.70
	1000	11.08	17.66	14.98	17.79	0.20	7.66	1.30	0.70

selected variables, as 20 uncorrelated causal factors for the model below to generate the responses. We then consider correlated causal factors in another simulation by replacing elements of A with 20 factors which are given by two randomly sampled non-overlapped series of 10 consecutive variables. The response Y was generated by the model

$$Y = \sum_{j \in A} \{\alpha_j I(Z_j = 0) + \beta_j I(Z_j = 1)\} + \varepsilon,$$

where $I(\cdot)$ was the indicator function, α_j and β_j were generated uniformly from $[-1.25, -0.75] \cup [0.75, 1.25]$, and the model error was a standard normal distribution. Sample size 300 is given for evaluation. For each setup, the proposed SPV algorithm, group lasso and SIS were applied to the same 200 simulated data sets. We used functions in the R package "grplasso" developed by Lukas Meier to produce group lasso results [13]. The SIS results are based on R package "sis". The average numbers of correctly and falsely identified factors in the 200 simulated data sets for both methods are summarized in Table 1. The lasso method is sensitive to the penalty parameter. For a naive comparison, we give results of group lasso with two fixed parameter values, denoted by $\text{Glasso}_{0.5}$ and $\text{Glasso}_{0.25}$ in Table 1, which correspond to 50% and 25% of the maximum penalty parameter estimates obtained from the lambdamax function in the R package. In the scenario of uncorrelated causal factor, the SPV algorithm with threshold values determined using the permutation approach with $\rho_1 = 0$ identified 19.93 of the 20 causal variables and 0.63 impostors on average for factor number $m = 200$. It is no surprise to see that $\text{Glasso}_{0.5}$ produced a smaller number of correct terms (11.8) and also a smaller number of false ones (0.15), because using a larger penalty parameter for the group lasso technique will identify fewer factors. While a smaller penalty parameter of $\text{Glasso}_{0.25}$ performed equally better in the identification of true factors (19.2), but more false factors (2.47). For the case of $m = 1000$, both SPV and $\text{Glasso}_{0.25}$ still performed very well in identifying correct factors, while SPV's performance was much better in terms of average false discovery numbers of 0.73 against 9.72 of $\text{Glasso}_{0.25}$. In the case of correlated causal factors, we have seen that SPV still performed better, although both SPV and group lasso lost some power in identifying correct factors. The R package "sis" allows users to determine the value of a parameter nsis which affects the performance.

Table 2 The average numbers of correctly identified QTL pairs from the SPV algorithm and R/qtlbim based on 100 repetitions with 10 sets of epistatic coefficients in the two-locus regression model. SPV_a indicates the threshold parameters are set using permutation approach with $\rho_1 = \rho_2 = a$

Epistatic coefficients	Number of correct sets			Number of false sets		
	R/qtlbim	SPV_0	SPV_2	R/qtlbim	SPV_0	SPV_2
Unif$(-1, 1)$	3.5	5.7	4.8	5.5	1.4	0.4
Unif$(-2, 2)$	5.4	8.4	8.2	3.3	1.6	0.3

We show the best results we obtained with $\texttt{nsis} = n/\log(n)$ in Table 1. The overall performance, SPV was better than SIS in this simulation setup.

Example 2 This example evaluates the feasibility of applying the SPV algorithm to the QTL mapping problem. We modified the classic two-QTL epistatic model [2] to have 10 sets of epistatic coefficients and no main effects. The model can be written as

$$y_i = \mu + \sum_{j=1}^{10} \{\beta_{1j} A_{ij_1} A_{ij_2} + \beta_{2j} A_{ij_1} D_{ij_2} + \beta_{3j} D_{ij_1} A_{ij_2} + \beta_{4j} D_{ij_1} D_{ij_2}\} + \varepsilon_i,$$

where y_i is the trait value, $\varepsilon_i \sim N(0, \sigma^2)$. The indicator variables $A_{ij_k} = \pm 1$ and $D_{ij_k} = 0.5$ or 0 are for the marginal additive and dominance effects, respectively, of QTL j_k. Hence, the interactions between the additive and dominance effects for QTL j_1 and j_2 are presented by multiplying together the respective additive and dominance indicator variables for QTL j_1 and j_2. In the simulations, 100 replicates of a progeny size of 500 from an F_2 cross experiment were simulated with 10 sets of the regression coefficients in the two-locus epistatic models using qb.sim.cross in the R package "qtlbim" [17]. The environmental variance σ^2 was fixed at 1 and μ was set at 10. The simulated genome consisted of 20 chromosomes, each of length 100 cM. The markers were evenly distributed in each chromosome with an interval size of 5 cM. The marker data were complete and without errors. For each repetition of the simulation, each of the 20 QTLs was located randomly at one of the positions $(0, 1, 2, \ldots, 99, 100$ cM) of a randomly selected chromosome.

This simulation study evaluates the ability of the SPV algorithm to identify marker pairs close to the true QTL pairs. Since R/qtlbim is one of the popular and powerful packages for QTL mapping, we applied it to the same simulated data for reference. In assessing the results, a QTL pair was considered to be correctly identified by the methods if the resulting marker pairs were within 5 cM of the true QTL pair. If an identified set of markers was not within 5 cM of any true QTL, that set of extraneous markers was then counted as a falsely identified set.

Two sets of 40 epistatic coefficients were generated uniformly from the two intervals, $(-1, 1)$ and $(-2, 2)$, respectively. Note that some of the generated coefficients may be close to zero. Table 2 summarizes the results of the SPV algorithm and the qb.scantwo method of R/qtlbim on 100 simulated data sets. In the R/qtlbim package, had we used the default setting for the parameters of the qb.scantwo function and

$2 \times \log(\text{Bayes factor}) = 2.1$ as the threshold for these simulated data sets, we would have tended to include too many false sets. To avoid this problem, we chose the 10 resulting pairs with the largest Bayes factors. For the scenario of 10 sets of epistatic coefficients from Unif$(-2, 2)$, the SPV algorithm with threshold values determined using the permutation approach with $\rho_1 = \rho_2 = 0$ correctly identified 8.4 of the 10 true pairs on average for each simulated data set, while the Bayesian method reported 5.4 true pairs. Our method also had fewer false positives: 1.6 pairs compared to 3.3 pairs for the Bayesian method. When we set the stringent parameter ρ_2 to 2, the average number of false sets was reduced to only 0.3, while the algorithm still correctly identified 8.2 of the 10 sets. For the other scenario of random epistatic coefficients, SVP performed consistently better than the Bayesian interval mapping method in identifying more correct QTL pairs and fewer falsely identified sets.

Example 3 To compare the SPV algorithm to the partition–retention method, we applied it to Example 5 of Chernoff et al. [4]. The example had $m = 1,000$ binary factor variables, denoted by $X_1, \ldots, X_{1000}$. The dependent variable Y was normally distributed with mean μ and standard deviation σ where

$$\mu = \max(\mu_1, \mu_2) + 0.1(\mu_1 + \mu_2) \quad \text{and} \quad \sigma = \max(\sigma_1, \sigma_2),$$

with $\mu_1 = 4X_1X_2X_3$, $\mu_2 = 6X_4X_5X_6X_7$, $\sigma_1 = 1 + X_1X_2X_3$, and $\sigma_2 = 1 + 2X_4X_5X_6X_7$. The binary explanatory variables were independent of each other and took on the value of 1 with probabilities 0.4, 0.5, 0.6, 0.35, 0.45, 0.55 and 0.65 for the seven influential variables. The probabilities for the remaining 993 variables were randomly selected from a uniform distribution in the range [0.4, 0.6].

We implemented the SPV algorithm to search for influential factors up to three-factor interactions for each simulated data set using threshold values corresponding to negative log of the 25th percentiles of Beta$(1, C_d^m)$, for $d = 1, 2, 3$. To evaluate the algorithm's performance, we defined correct identification of the two sets of influential variables as follows. We determined X_1, X_2, and X_3 to be correctly uncovered if they could be linked together by the identified pairs or triplet. Similarly, if X_4, X_5, X_6 and X_7 could be linked together by the identified pairs or triplets, we determined the second set of these four influential variables to be correctly uncovered. Any identified sets containing variables other than the seven causal ones were counted as false sets. Based on 600 simulated data sets of sample size 400, the SPV revealed the first set 99%, and the second set 55%. The percentages of seven individual variables appeared in the identified sets are 100% for the first four variables, 99% for X_5, 91% for X_6, and 62% for X_7. In fact, the SPV algorithm revealed all seven influential variables in 60% of the 600 data sets, and the average number of falsely identified variables was only 1.2 out of 993. These results were better than those of Chernoff et al. [4], which complex analysis reported that the seven influential variables were completely uncovered in only one of five simulated data sets and much large number of falsely identified variables.

4 Real Data Analysis

QTL Study of High Density Lipoprotein Cholesterol

We applied the SPV algorithm to the data set of the QTL study of high density lipoprotein (HDL) cholesterol from Ishimori et al. [9]. One of the objectives was to identify loci controlling the plasma HDL levels. In this experiment, C57BL/6J (B6) and 129S1/SvImJ (129) mice were mated to produce the (B6 $\times$ 129) F_1 progeny, which were interbred to produce 294 female F_2 progeny. Female B6 mice have low plasma HDL levels and are susceptible to atherosclerosis; in contrast, female 129 mice have high plasma HDL levels. The plasma HDL concentrations and genotypes of 113 marker of the 294 F_2 progeny are available from the QTL archive at the Jackson Laboratory website, http://cgd.jax.org/nav/qtlarchive1.htm. We excluded two mice with missing HDL levels. The SPV algorithm with threshold values determined using the permutation approach with parameter $\rho_d = 0$ identified four influential single markers with locations Chr1@101.2, Chr12@22, Chr9@26 and Chr8@43, shown in Table 3. These markers are close to what was reported by Ishimori et al. [9], where analyses were carried out using a Bayesian method proposed by Sen and Churchill [14]. While the SPV algorithm revealed no significant pairs, it found an influential triplet (Chr1@81.6, Chr1@109, Chr2@105) that contains two pairs close to the reported pairs, (Chr1@80, Chr1@104) and (Chr1@104, Chr2@90), in the same literature.

We used the package R/qtl [1] to carry out multiple-regression analysis on the identified markers. The results summarized in Table 3 confirm that all of the identified terms were significant. The identified seven markers accounted for 47.7% of the variance in HDL levels. The most important marker Chr1@109 explained 17.5% of the total variance, which is a little larger than that of Chr1@104 reported in Ishimori et al. [9]. Overall the two analyses produced almost the same results, except that

Table 3 Multiple-regression analysis of variance for log HDL in 292 (B6 $\times$ 129) F_2 females. The last column lists the QTL locations identified by Ishimori et al. [9] that are close to the markers identified by the SPV algorithm

Location (marker name)	df	*SS*	%Var	p-value	Ishimori et al.
Chr1@101.2 (D1MIT406)	2	0.156	2.656	0.0017	
Chr12@22 (D12MIT172)	2	0.172	2.929	0.0009	Chr12@20
Chr9@26 (D9MIT129)	2	0.164	2.792	0.0013	Chr9@24
Chr8@43 (D8MIT248)	2	0.132	2.240	0.0046	Chr8@44
Chr1@81.6 (D1MIT159)	18	0.748	12.731	0.0000	Chr1@80
Chr1@109 (D1MIT210)	18	1.031	17.543	0.0000	Chr1@104
Chr2@105 (D2MIT148)	18	0.821	13.960	0.0000	Chr2@90
Chr1@81.6:Chr1@109	12	0.622	10.586	0.0000	Chr1@80:Chr1@104
Chr1@81.6:Chr2@105	12	0.450	7.649	0.0004	
Chr1@109:Chr2@105	12	0.678	11.539	0.0000	Chr1@104:Chr2@90
Chr1@81.6:Chr1@109:Chr2@105	8	0.224	3.814	0.0191	
Total	291	5.879	47.676		

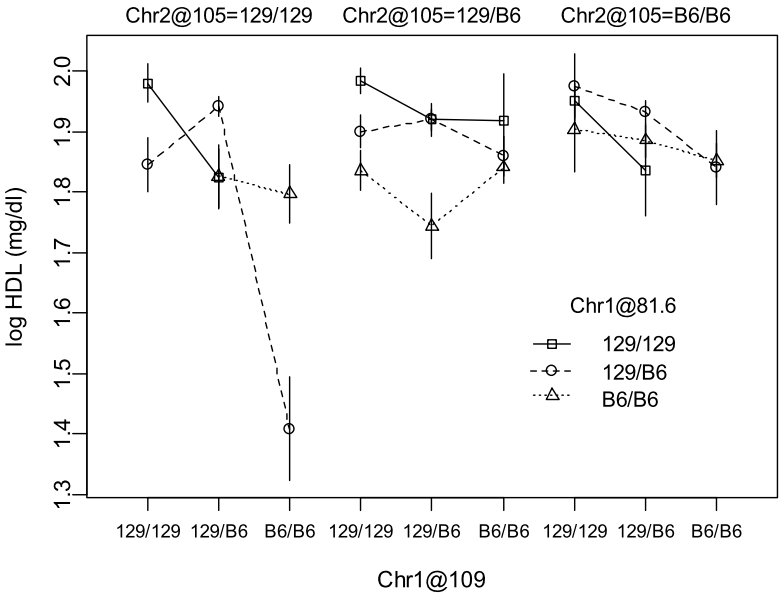

Fig. 4 The effects of interactions between three identified markers contributing to changes in plasma HDL concentrations. Mean values of log transformed HDL with one standard error are represented for all the nonempty combinations

the SPV algorithm identified an extra triplet. We also computed the cell means and standard deviations of log HDL levels for the 27 genotype combinations of the three markers, and plot them in Fig. 4.

The combination of homozygous B6 alleles at Chr1@109, homozygous 129 alleles at Chr2@105, and heterozygous alleles at Chr1@81.6 led to extremely low HDL concentrations. Although there were only five mice in this category, their log HDL levels, 1.62, 1.53, 1.48, 1.2, and 1.2, fall below the 5th percentile of the 292 observations. In fact, four of them were in the long tail of the distribution of these log HDL observations, of which 1.2 is actually the smallest value. Since low HDL levels relate to the occurrence of atherosclerosis, the combination of these three markers may be a genetic predictor for the risk. It may also be worth further exploring any candidate genes for this triplet.

Genome-Wide Loci Search for ACE Activity

We demonstrated the proposed SPV algorithm using a recent two-stage genome-wide association study (GWAS) which aimed to identify quantitative trait loci causally related to an intermediate phenotype: the activity of Angiotensin-Converting Enzyme (ACE), a key enzyme of the well-known renin–angiotensin–aldosterone system that is pivotal for electrolyte balance and blood pressure regulation [5]. A total of 1,023 hypertensive subjects were recruited by the Academia Sinica Multi-Centered Young-Onset Hypertension (AS-YOH) Genetic Study. The study included a total of 400 hypertensive subjects at the initial GWAS stage, and 623 hypertensive subjects in the

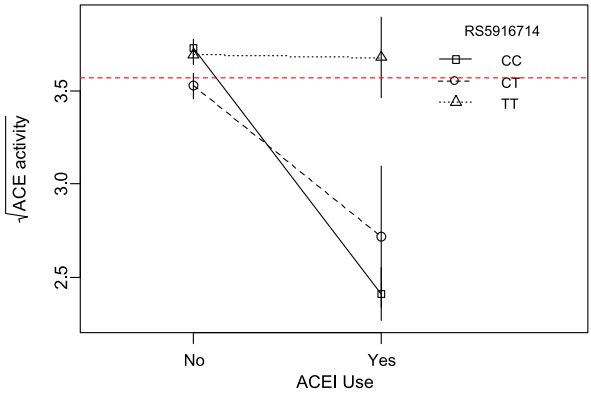

Fig. 5 The effects of interaction between ACEI intake and SNP rs5916714 on changes in ACE activity (UI^{-1}). Mean values of square root of ACE activity with one SE are represented for each of the six combinations. *The horizontal line* is the overall mean of the square root of ACE activity

confirmatory stage. Genotyping experiments were performed by the deCODE genetics with 400 leukocyte DNA samples, using the Illumina Infinium II Human-Hap550 SNP. Corresponding to each of the 400 subjects are 560,159 SNPs of three genotypes. Chung et al. set a genome-wide level of significance of 10^{-7} for multiple testing corrections in the first stage, and screened out eight SNPs. In the second stage, they genotyped an additional 623 AS-YOH subjects for the eight identified SNPs. They used a stepwise linear model to fit ACE activities with the eight SNPs for the 1,023 subjects, and reported in the second stage that three SNPs were significantly associated with ACE activity: rs4343 in the ACE gene ($p = 3.0 \times 10^{-25}$), and rs495828 ($p = 3.5 \times 10^{-8}$) and rs8176746 ($p = 9.3 \times 10^{-5}$) in the ABO gene.

We applied the SPV algorithm to the 400 subjects' ACE activity measurements using $m = 560,159$ SNPs. Two binary factors, gender and use of the ACE Inhibitor (ACEI), were also included. The first run detected rs4343 as an important factor with a $-\log$ p-value equal to 50.62, far above the threshold of 14.48, –log of the 25th percentile of Beta$(1, m)$. The second run identified the use of ACEI, with a $-\log$ p-value equal to 37.8. The SNP rs495828 was found in the third run, with a $-\log$ p-value of 18.83, and the screening for single factor stopped at the next run. We proceeded to screen all pairs using a threshold of 27.03 corresponding to –log of the 50th percentile of Beta$(1, C_2^m)$. The screening identified the pair of ACEI use and rs5916714, with $-\log$ p-value 27.99–slightly larger than the loose threshold. We computed the cell means and standard deviations of the square root of ACE activity for the 6 combinations of the SNP rs5916714 and ACEI use, and plot them in Fig. 5. Figure 5 shows that genotype TT of the SNP rs5916714 had no expected effect on inhibiting ACE activity among ACEI users. In summary, we have exactly identified the two important SNPs in ACE and ABO genes significantly associated with ACE activity as reported in Chung et al. [5]. The revealed interaction of SNP rs5916714 and ACEI intake, albeit marginal, is nonetheless interesting. If it is true, we have added important pharmacogenetic knowledge on ACEI. Although it may be a false finding for near the threshold, the finding merits further evaluation using independent samples.

5 Discussion and Conclusion

We have proposed an algorithm to identify influential single factors and multi-factor interactions related to a continuous trait variable. The main idea of the proposed SPV algorithm is to stepwise pare down the total variation of responses so that the remaining influential factors have increased chances of being identified. For the demonstration of the potential of SPV algorithm, we have used simulation schemes similar to those proposed in the literature on the group lasso method, SIS, Bayesian QTL mapping and the partition–retention method to generate various data sets that are supposed to be favorable to these competitive methods. Although the SPV algorithm seemed to perform better than these competitive methods in the simulation studies with better identification power and smaller false discovery rate, we could only claim that the SPV algorithm is a potential method for variable screening. With the R package "spv", available at http://www.stat.sinica.edu.tw/jshwang, users could simply implement the screening package with the default threshold value of the 25th percentile of Beta$(1, C_d^m)$. Although a threshold will affect false negatives and positives, it is relatively easy to explore. One may rerun SPV by setting a stringent threshold with 5th percentile when reducing false positives is needed or a higher 50th percentile to avoid losing important factors and allow some unimportant factors.

Running time is always a concern for algorithms involving exhaustive screening. Like other popular methods, the burden of computation for the SPV algorithm may hinder its direct application to searches for high-order interactions in cases consisting of more than hundreds of thousands of factors. It is feasible in practice, however, for data sets of moderate size. We have successfully implemented the SPV algorithm to screen about 500,000 SNPs for influential two-factor interactions in three days using a high-level PC cluster, but it is not feasible to proceed to screen for higher-order interactions. One possible solution is to thin the candidate factors from stage to stage [4]. The idea comes from the fact that marginal effects of a set of influential factors tend to be larger than those of noisy factors. Hence, we can discard those factors with extremely weak marginal effects to form a smaller subset of candidate variables for the next higher-order interaction screening.

The SPV algorithm is very similar to the conventional stepwise forward regression. Although the slight modification has made SPV feasible for dealing with interactions in ultrahigh-dimensional data, it may lose some power as shown in Example 1 when the causal factors are correlated. The major difference between these two methods is the stopping rule which is critical to the performance of screening. The proposed percentile of Beta$(1, C_d^m)$ as a universal threshold for d-dimensional screening is very different from the conventional rule of using AIC or BIC criterion. Wang [15] has investigated forward regression for ultrahigh-dimensional variable screening using BIC criterion of Chen and Chen [3]. It may be worth making a comparison study. But we have not gotten a chance to do it because R package of Wang's method is not publicly available and its focus is on screening single factors. Furthermore, there may be some other competitive methods too. We acknowledge that each method has its advantage and limitation. In this study, we only conclude that the simple SPV algorithm is a potential alternative choice for its great performances in the comparison with several popular methods. Besides, we have demonstrated that the SPV algorithm could be applied to screen interactions in ultrahigh-dimensional data in practical world.

Acknowledgements The authors are grateful to an AE and two referees for the helpful comments and valuable suggestions. The authors thank Professor W.H. Pan of Academia Sinica for providing the GWAS data and the permission of use as a real example in this study. The work was supported in part by the National Science Council of Taiwan grant NSC 96-2118-M-001-003.

References

1. Broman KW, Wu H, Sen S, Churchill GA (2003) R/qtl: QTL mapping in experimental crosses. Bioinformatics 19:889–890
2. Carlborg Ö, Andersson L, Kringhorn B (2000) The use of a genetic algorithm for simultaneous mapping of multiple interacting quantitative trait loci. Genetics 155:2003–2010
3. Chen J, Chen Z (2008) Extended Bayesian information criterion for model selection with model spaces. Biometrika 95:759–771
4. Chernoff H, Lo SH, Zheng T (2009) Discovering influential variables: A method of partitions. Ann Appl Stat 3:1335–1369
5. Chung CM, Wang RY et al (2010) A genome-wide association study identifies new loci for ACE activity: potential implications for response to ACE inhibitor. Pharmacogenomics J 10(6):537–544
6. Fan J, Lv J (2008) Sure independence screening for ultra-high dimensional feature space (with discussion). J R Stat Soc B 70:849–911
7. Fan J, Lv J (2010) A selective overview of variable selection in high dimensional feature space. Stat Sin 20:101–148
8. David HA (1980) Order statistics. Wiley, New York
9. Ishimori N, Li R et al (2004) Quantitative trait loci analysis for plasma HDL-cholesterol concentrations and atherosclerosis susceptibility between inbred mouse strains C57BL/6J and 129S1/SvImJ. Arterioscler Thromb Vasc Biol 24:161–166
10. Liang Y, Kelemen A (2008) Statistical advances and challenges for analyzing correlated high dimensional SNP data in genomic study for complex diseases. Stat Surv 2:43–60
11. Lo SH, Chernoff H, Cong L, Ding Y, Zheng T (2008) Discovering interactions among BRCA1 and other candidate genes associated with sporadic breast cancer. Proc Natl Acad Sci USA 105(34):12387–12392
12. Loughin TM (2004) A systematic comparison of methods for combining p-values from independent tests. Comput Stat Data Anal 47:467–485
13. Meier L, van de Geer S, Bühlmann P (2008) The group Lasso for logistic regression. J R Stat Soc B 70:53–71
14. Sen S, Churchill GA (2001) A statistical framework for quantitative trait mapping. Genetics 159:371–387
15. Wang H (2009) Forward regression for ultra-high dimensional variable screening. J Am Stat Assoc 104:1512–1524
16. Wu TT, Chen YF, Hastie T, Sobel E, Lange K (2009) Genome-wide association analysis by lasso penalized logistic regression. Bioinformatics 25:714–721
17. Yandell BS, Mehta T et al (2007) R/qtlbim: QTL with Bayesian interval mapping in experimental crosses. Bioinformatics 23:641–643
18. Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. J R Stat Soc B 68:49–67